

Решения

Задача 1.

1 вариант.

Есть два способа получить требуемый объём.

1 способ.

Наливать в ведро **А**, переливать в **Б**, опустошать **Б**, если оно полно.

Последовательно выполняя эти действия, будут получаться следующие объёмы в вёдрах:

Действие		Б	А	00	А	Б	А	00	А	00	А	Б	А	00	А
Ведро А	0	0	7	0	5	5	7	0	7	0	3	3	7	0	7
Ведро Б	0	12	5	5	0	12	10	10	3	3	0	12	8	8	1

Где действие Б обозначает наполнить ведро **Б**.

Действие А – пересыпать ведро **Б** в ведро **А**.

Действие 00 – освободить ведро **А**.

Таким образом, на двенадцатом действии в ведре **Б** получаем необходимый объём. Для чего потребовалось 7 наполнения ведра **А**, из которых только 5 наполнения о краёв.

Задача 2.

1 вариант.

Необходимо лишь в цикле уменьшать М на требуемое число орешков, пока это возможно.

Пример программы(на C++).

```
#include <iostream>
using namespace std;
int main(){
    int n,m,k;
    cin>>n>>m;
    k=1;    //первому даётся один орешек
    while(m>=k && k<=n){ //пока орешков хватает и круг не
        пройден - раздаём
        m-=k;
        ++k;    //следующему на один больше
    }
    cout<<m;    //оставшееся число орешков

    return 0;
}
```

2 вариант.

Решение аналогично.

Пример программы(на C++).

```
#include <iostream>
using namespace std;
int main(){
    int n,m,k;
    cin>>n>>m;
    k=3;    //первому даётся три орешка
```

```

while(m>=k && k<=n){ //пока орешков хватает и круг не
пройден - раздаём
    m-=k;
    k+=2; //следующему на два больше
}
cout<<m; //оставшееся число орешков

return 0;
}

```

Задача 3.

1 вариант.

Если кентавр стоит на чёрной клетке, он ходит, как слон. А слон ходит всегда только по клеткам одного цвета, значит, на белую клетку он никак не попадёт. А у чёрной клетки все соседние белого цвета. Следовательно, с чёрной клетки кентавр не может попасть на соседнюю клетку.

Таким образом, кентавр не всегда может попасть на соседнее поле.

2 вариант.

Если всадник стоит на чёрной клетке, он ходит как ладья. Тогда он не может попасть на соседнее поле, так как он должен ходить не менее, чем на два хода.

Таким образом, всадник всегда может попасть на соседнее поле.

Задача 4.

1 вариант.

Для каждой белочки в очереди можно просто просматривать всех белочек от неё до начала очереди и искать на этом отрезке белочку с наибольшим количеством орешков, т.е. максимум.

2 вариант.

Решение аналогично, только нужно искать самую молодую белочку на том же отрезке, т.е. минимум.

Пример решения(на C++).оба варианта

```

#include <iostream>
using namespace std;
int main(){
    int n,mass[10000],num,index;

```

```

cin>>n;
for(int i=0;i<n;++i)
    cin>>mass[i];
for(int i=0;i<n;++i){
    num=mass[i];
    index=i;
    for(int j=i+1;j<n;++j){
        if(mass[j]>num){ //ищем максимум(для второго варианта
заменить
            num=mass[j]; // знак на <, для поиска минимума)
            index=j;
        }
    }
    cout<<index-i-1<<" "; //расстояние до максимума(минимума)
}

return 0;
}

```

Задача 5.

1 вариант.

В примере указано, как «переместить» число из одной ячейки в другую. Иначе можно описать этот алгоритм так: пока в первой ячейке не ноль, число в первой ячейке уменьшаем на 1 и число во второй ячейке увеличиваем. Таким образом, a раз уменьшается, число в первой ячейке, и a раз увеличивается число во второй ячейке.

Вернёмся к поставленной задаче. Необходимо a раз прибавить b в ячейку содержащую 0. Так мы получим число $a \cdot b$, которое потом легко переместить в первую ячейку. Уменьшая a на 1, мы должны увеличить на b ячейку номер 3 или 4, будем использовать ячейку номер 4 для хранения результата. Ячейка номер 3 понадобится для другого. Уменьшая b мы можем потерять значение этого числа. Чтобы не допустить этого, будем в момент уменьшения b , увеличивать значение в ячейке номер 3, тем самым сохраняя значение числа b . Которое затем обратно переместим во вторую ячейку, для следующего шага цикла по числу a .

Пример программы.

1. D 1 (1-11 основной цикл)
2. T 12

3. D 2 (3-7 перемещение из ячейки №2 в ячейку №3 и прибавление в ячейку №4)
4. T 8
5. I 3
6. I 4
7. T 3
8. D 3 (8-11 перемещение из ячейки №3 в ячейку №2)
9. T 1
10. I 2
11. T 8
12. D 4 (12-14 перемещение значения из ячейки номер 4 в ячейку номер 1)
13. T 0
14. I 1
15. T 12

2 вариант.

Решение аналогично. Нужно лишь вторую команду T 12 заменить на T 0, т.к. искомое число уже будет в ячейке №4, и убрать лишние команды 12-15.